PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	09/738,349
Filing Date	12/14/2000
First Named Inventor	Elizabeth Adleberg Brodsky
Art Unit	2153
Examiner Name	Yasin M. Barqadle
Attorney Docket Number	AUS920000510US1
Total Number of Pages in This Submission	25

ENCLOSURES *(Check all that apply)*

☑ Fee Transmittal Form
  ☑ Fee Attached
☐ Amendment/Reply
  ☐ After Final
  ☐ Affidavits/declaration(s)
☐ Extension of Time Request
☐ Express Abandonment Request
☐ Information Disclosure Statement
☐ Certified Copy of Priority Document(s)
☐ Reply to Missing Parts/ Incomplete Application
  ☐ Reply to Missing Parts under 37 CFR 1.52 or 1.53

☐ Drawing(s)
☐ Licensing-related Papers
☐ Petition
☐ Petition to Convert to a Provisional Application
☐ Power of Attorney, Revocation Change of Correspondence Address
☐ Terminal Disclaimer
☐ Request for Refund
☐ CD, Number of CD(s) ____________
  ☐ Landscape Table on CD

☐ After Allowance Communication to TC
☐ Appeal Communication to Board of Appeals and Interferences
☑ Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
☐ Proprietary Information
☐ Status Letter
☐ Other Enclosure(s) (please identify below):

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Law Office of Anthony England		
Signature	Anthony V S England		
Printed name	Anthony V.S. England		
Date	6-2-2006	Reg. No.	35,129

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature	Anthony V S England		
Typed or printed name	Anthony V.S. England	Date	6-2-2006

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: **Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/17i (11-04)
Approved for use through 07/31/2007. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PROCESSING FEE
Under 37 CFR 1.17(i)
TRANSMITTAL

(Fees are subject to annual revision)

Send completed form to: Commissioner for Patents
P.O. Box 1450, Alexandria, VA 22313-1450

Application Number	09/736,349
Filing Date	12/14/2000
First Named Inventor	Elizabeth Adleberg Brodsky
Art Unit	2153
Examiner Name	Yasin M. Barqadle
Attorney Docket Number	AUS920000510US1

Enclosed is a paper filed under 37 CFR 41.20(b)(2) that requires a processing fee (37 CFR 1.17(i)).
Payment of $ 500.00 is enclosed.
This form should be included with the above-mentioned paper and faxed or mailed to the Office using the appropriate Mail Stop, if applicable. *For transmittal of petition fees under 37 CFR 1.17(f), (g) or (h), see form PTO/SB/17p.*

Payment of Fees (small entity amounts are NOT available for the petition fees)

☒ The Commissioner is hereby authorized to charge the following fees to Deposit Account No. 09-0447 :
☒ processing fee under 37 CFR 1.17(i) ☒ any deficiency of fees and credit of any overpayments
Enclose a duplicative copy of this form for fee processing.

☐ Check in the amount of $ ________ is enclosed.

☐ Payment by credit card (Form PTO-2038 or equivalent enclosed). Do not provide credit card information on this form.

Processing Fees under 37 CFR 1.17(i): Fee $130 Fee Code 1808 for all,
Except for § 1.221 papers (Fee Code 1803)

For papers filed under:
§ 1.28(c)(3) - for processing a non-itemized fee deficiency based on an error in small entity status.
§ 1.41 - for supplying the name or names of the inventor or inventors after the filing date without an oath or declaration as prescribed by § 1.63, except in provisional applications.
§ 1.48 - for correcting inventorship, except in provisional applications.
§ 1.52(d) - for processing a nonprovisional application filed with a specification in a language other than English.
§ 1.53(b)(3) - to convert a provisional application filed under § 1.53(c) into a nonprovisional application under § 1.53(b).
§ 1.55 - for entry of late priority papers.
§ 1.99(e) - for processing a belated submission under § 1.99.
§ 1.103(b) - for requesting limited suspension of action, continued prosecution application (§ 1.53(d)).
§ 1.103(c) - for requesting limited suspension of action, request for continued examination (§ 1.114).
§ 1.103(d) - for requesting deferred examination of an application.
§ 1.217 - for processing a redacted copy of a paper submitted in the file of an application in which a redacted copy was submitted for the patent application publication.
§ 1.221 - for requesting voluntary publication or republication of an application. **Fee Code 1803**
§ 1.291(c)(5) - for processing a second or subsequent protest by the same real party in interest.
§ 1.497(d) - for filing an oath or declaration pursuant to 35 U.S.C. 371(c)(4) naming an inventive entity different from the inventive entity set forth in the international stage.
§ 3.81 - for a patent to issue to assignee, assignment submitted after payment of the issue fee.

Anthony V S England
Signature

6-2-2006
Date

Anthony V.S. England
Typed or printed name

35,129
Registration No., if applicable

This collection of information is required by 37 CFR 1.17. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 5 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

In the United States Patent and Trademark Office

In re the application of: Brodsky	
Filed: 12/14/2000	Group Art Unit: 2153
For: Method Apparatus and Computer Program Product to Crawl a Web Site	Examiner: Scott M. Klinger
Application No. 09/736,349	
Applicant's Docket: AUS920000510US1	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Dear Sir:

REAL PARTY IN INTEREST

The assignee, International Business Machines Corporation, is the real party in interest.

RELATED APPEALS AND INTERFERENCES

This is the first appeal in the present patent application. There are no other appeals or interferences known to the appellant or its legal representative. International Business Machines Corporation is the sole assignee of the patent application.

STATUS OF CLAIMS

Claims 1-7, 9-15, and 17-23 are pending in the application. All the pending claims stand rejected. Fourth Office action (the "Final Office Action"), January 4, 2006. Appellant has timely appealed from the final rejection. Notice of Appeal, sent by facsimile transmission to the USPTO on April 4, 2006. This case has never been appealed or subject to continued examination.

The claims appealed herein, and for which arguments are herein presented, are claims Claims 1, 2, 4, 7, 9, 10, 12, 15, 17, 18, 20 and 23. [1]

History of the Case

A first Office action, mailed July 19, 2004, rejected all claims and relied on three references, U.S. Patent No. 6,301,614 ("Najork"), U.S. Patent No. 6,026,413 ("Challenger"), and U.S. Patent No. 6,748,418 ("Yoshida"). Reply A was filed October 18, 2004, amending Claims 1, 2, 4, 5, 7, 9, 10, 12, 13, 15, 17, 18, 20, 21, and 23. Claims 8, 16, and 24 were canceled.

A second, final Office Action of February 24, 2005, rejected all remaining claims and cited additional references, "Crawler-Friendly Web Servers" September 2000, ("Brandman") and U.S. Patent No. 6,735,169 ("Albert"). In a first Request for Reconsideration, filed March 21, 2005, Appellant requested that the finality of the second Office action be withdrawn because Appellant was not granted an interview before the final rejection and because the statements of rejections of claims 7, 15 and 23 in the second Office action were identical repetition of the rejections of the first Office action, even though these claims were substantially amended in Reply A.

On March 22, Examiner and Attorney for Appellant discussed the application in a telephone conference.

A third, final Office Action, mailed on June 22, 2005, withdrew the Brandman reference relied upon in the second Office action and finally rejected all claims in reliance

[1] Arguments are *not* herein presented regarding claims 3, 5, 6, 11, 13, 14, 19, 21 and 22. However, Appellant contends, of course, that these claims are allowable since they depend on claims for which arguments are herein presented and which Appellant contends are allowable. MPEP 2143.03 (citing *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988).

upon Najork, Challenger, Albert and a new reference, U.S. Patent No. 6,665,658 ("DaCosta"). In a Statement of Disqualification of Reference and Request for Reconsideration, filed August 11, 2005, Appellant provided facts disqualifying the DaCosta reference and requested allowance.

The present, fourth Office action (the "Final Office Action"), dated January 4, 2006, withdrew the DaCosta reference and, once again, finally rejected all claims, citing a new reference, US Patent 6,449,636 ("Kredo").

STATUS OF AMENDMENTS

There are no amendments in connection with this appeal and none were submitted subsequent to the Final Office Action. The claims in the Claim Appendix herein set out the claims as previously amended, i.e., prior to the Final Office Action.

SUMMARY OF CLAIMED SUBJECT MATTER

The claims in the present case relate to an embodiment of the invention in which a reference to a web page that is not simply set out on the page as a hyperlink address, but is instead specified by a script, may be produced only when a client browser executes the reference.[2] Present application, page 5, lines 10 - 15. See also, page 12, line 21- page 13, line 1 (describing how the reference may be specified by a script, a selection menu, form, button or other similar element). This presents a problem for a web page crawler. According the present application, the browser locates code for the function that is called and then executes the specified function using an applet.

Claims 1, 9 and 17

Claim 1 describes a method for crawling a web site. The crawler 171 is programmable to perform particular action sequences 305 for generating queries to the web server 100. See present application, FIG's 1 and 3, page 14, line 10 - page 15, line 2. At least one page of the web site has a reference specified by a script 303 to produce an address for a next page, so that the address is produced only when a client browser 205 executes the reference. See present application, FIG's 1, 2, and 3, page 15, lines 2-8. The crawler 171 parses a reference from one of the web pages and sends the reference to an applet running in a browser 205. See present

[2] The broad claims in this case say a reference is *specified* by a script because an href tag, for example, typically has a *call* to a script and not the script *itself*.

application, FIG. 2, page 12, lines 17- page 13, line 3. See also FIG. 5 (step 510) and page 16, lines 16-17. The browser 205 determines the address for the next page by executing the reference and sending the address to the crawler 171. Present application, FIG. 2, page 13, lines 3-8. See also FIG. 5 (steps 515 and 500) and page 16, lines 17-19.

Claims 9 and 17 have similar language. Claim 17 is an apparatus form of claim 1, including a processor connected to a network. See present application, page 7, line 22. A storage device is connected to the processor and the network for storing a program for controlling the processor. See present application, page 7, lines 22-23.

Claims 2, 10 and 18

Claim 2 describes an embodiment of claim 1 in which the browser 205 is configured to use a proxy 215 and refer to a resolver file 405 for hostname-to-IP-address-resolution. See present application, FIG. 4, page 15, lines 15-17, see also FIG. 5 (steps 525 and 530) and page 16, lines 19-20. The web site server 100 has an IP address and the proxy 215 for the browser 205 has a certain IP address. See present application, FIG. 4, page 6, lines 8-9, page 15, lines 17-20. The IP address of the proxy 215 is different than the IP address of the web site server 100, and the resolver file indicates the IP address of the proxy 215 as the IP address for the web site server 100. See present application, FIG. 4, page 6, lines 9-14, page 15, line 20 - page 16, line 2.

Claims 10 and 18 have similar language.

Claims 4, 12 and 20

Claim 4 states that the crawler 171 is programmable to perform particular action sequences for selecting non-hypertext-link parameters from at least one web page in a particular sequence. See present application, page 14, line 10-11. Queries to the web server 100 include the selected parameters and a context arising from the particular sequence. See present application, FIG. 3, page 14, line 12 - page 15, line 8.

Claims 12 and 20 have similar language.

Claims 7, 15 and 23

Claim 7 describes a method for reducing dynamic data generation on a web site server 100. A web site server 100 is queried 250 by a crawler program 171 responsive to references from one web page to another in the web site, wherein the queries 250 are for causing the server 100 to generate web pages and at least some of the web pages are dynamically generated. See present application, FIG. 2, page 13, lines 16-19. The server 100 generated web pages are processed to generate corresponding processed versions of the web pages that can be served in response to future queries, reducing dynamic generation of web pages by the server 100. See present application, FIG. 2, page 13, lines 19-23 . At least a first such server generated web page has included in it an operation that would cause the server 100 to dynamically generate a second web page if the first page were used to generate further requests to the server 100. See present application, FIG. 5 (STEPS 505-520) and page 16, lines 16-19 . The processing of the server generated web pages includes removing the operation from the first server generated web page and replacing the operation with a reference to a version of another of the server generated web pages. See present application, FIG. 5, page 16, line 23 -n page 17, line 4 (step 545).

Claims 15 and 23 have similar language. Claim 23 is an apparatus form of claim 1, including a processor connected to a network. See present application, page 7, line 22. A storage device is connected to the processor and the network for storing a program for controlling the processor. See present application, page 7, lines 22-23.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

First ground of rejection for review: Independent claims 1, 9 and 17 stand rejected under 35 U.S.C. 103(a) in view of the combination of U.S. Patent No. 6,301,614 ("Najork") and US Patent 6,449,636 ("Kredo"). Final Office Action, Section 4. Appellant respectfully submits the rejection is improper.

Second ground of rejection for review: Dependent claims 2, 10 and 18 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Najork, Kredo and US Patent 6,735,169 Albert. Final Office Action, section 5. Appellant respectfully submits the rejection is improper.

Third ground of rejection for review: Dependent claims 4, 12 and 20 are mentioned in section 4 of the Final Office Action but never otherwise discussed in that Action. In this Appeal Brief, it is assumed, based on previous Office actions, that it was the Examiner's intent to reject these claims in the Final Office Action in view of a combination of asserted teachings by Challenger, Najork, and Kredo and that it was the Examiner's intent to discuss these claims in connection with dependent claims 5-6, 13-14 and 21-22 in section 6 of the Final Office Action. Appellant respectfully submits the assumed rejection is improper.

Fourth ground of rejection for review: Independent claims 7, 15 and 23 and dependent claims 5-6, 13-14 and 21-22 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Najork, Kredo and Challenger. Final Office Action, sections 6 and 7. Appellant respectfully submits the rejection is improper.

ARGUMENTS

First ground of rejection for review: Claims 1, 9 and 17

The subject claims state that a page has a reference and that the reference is specified by a script for producing an address for a next page. The claims go on to say that such a reference is parsed from one of the web pages by the crawler program and sent to an applet running in a browser. Further, the claims clearly tie these aspects together by stating that the address for the next page is determined by the browser "executing" the reference and sending the address to the crawler. Appellant respectfully submits that all the limitations of the subject claims are not taught or suggested by the art relied upon and that the rejection is, therefore, improper. MPEP 2143.03 (citing *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974); *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970)).

The references do not teach or suggest parsing a reference from one of the web pages by a crawler program and sending the reference to an applet running in a browser.

The Final Office Action maintains Najork teaches that a method of web crawling includes "parsing such a reference from one of the web pages by the crawler program and sending the reference to an applet running in a browser," as in the present claim 1. (Claims 9 and 17 have similar language.) However, Najork does not teach parsing *a reference* (such as a reference having that a script for producing an address for a next page) and sending the reference *to an applet running in a browser*, as claimed. Najork teaches parsing a URL by a crawler and sending the parsed URL somewhere for compiling URL's, not to an applet running in a browser.

More specifically, Najork describes the problem it addresses as follows:

> A web crawler is a program that automatically finds and downloads documents from host computers in an Intranet or the world wide web . . . Before the web crawler downloads the documents associated with the newly discovered URL's, the web crawler needs to find out whether these documents have already been downloaded . . . Thus, web crawlers need efficient data structures to keep track of downloaded documents and any discovered addresses of documents to be downloaded. Such data structures are needed to facilitate fast data checking and to avoid downloading a document multiple times.

Najork, col. 1, lines 34-61. The teachings of Najork concern "the data structures and methods used to keep track of the URL's of documents that have already been downloaded or that have already been scheduled for downloading." Najork, col. 4, lines 54-57.

The web crawler taught by Najork includes "threads 130 for downloading web pages from the servers 112, and processing the downloaded web pages; a main web crawler procedure 140 executed by each of the threads 130; and a URL processing procedure 142 executed by each of the threads 130 to process the URL's identified in a downloaded web page." Najork, col. 3, lines 31-58. Each thread executes a main web crawler procedure 140 shown in FIG. 3. Najork, col. 4, lines 58-59. The web crawler thread determines the URL of the next document to be downloaded (step 160) and then downloads the document corresponding to the URL, and processes the document (162). Najork, col. 4, lines 59-64. According to that processing, the main procedure identifies URL's in the downloaded document that are candidates for downloading and processing (step 162). Najork, col. 4, line 66 - col. 5, line 3.

Najork specifically points out that "these URL's are typically found in hypertext links in the document being processed." Najork, col. 5, lines 3-4. But, as particularly pointed out in the present application, sometimes URL's are *not* found in hypertext links, which presents a problem. That is, references from one web page to another may not be simply set out on the page as a straightforward hyperlink address [i.e., a URL], instead may be a script, form, selection menu, or button for example. Consequently, a conventional crawler and the crawler taught by Najork are not suitable for the "staticizing" problem addressed in the present invention. Present application, page 2, line 20 - page 3, line 6 (concluding, "Thus a need exists for improvements in crawler programs, to overcome their limitations so that they may be used for the staticizing problem as well as other applications."). Najork offers no teaching that addresses this problem, or even that suggests it exists.

The present application further elaborates on the problem, explaining how a reference that is not "simply set out on the page as a hyperlink address, but instead . . . specified by a script, for example, so that the address is produced only when a client browser executes the reference." Present application, page 5, lines 10 - 15; see also, page 12, line 21- page 13, line 1 (describing how the reference may be specified by a script, a selection menu, form, button or other element).

The present application goes on to explain how this problem may be addressed, as follows:

> To generate references of this sort in connection with generating the requests to the server, another aspect of the invention arises. According to an embodiment, the crawler parses each received web page and sends references to an applet developed for an embodiment of the present invention that runs in the browser. (This applet may be referred to herein as a "JavaScript execution engine" or simply "JEE.") The browser determines the address for a next page responsive to such a reference, so that the browser may receive the next page and any cookie for the next page from the server, and the JEE returns the address and any cookie to the crawler program.

Present application, page 5, lines 15-22; page 15, lines 2-8. Accordingly, the subject claims state that the invention includes parsing a reference from one of the web pages by a crawler program and sending the reference to an applet running in a browser. The references do not teach or suggest this.

The references do not teach or suggest determining the address for the next page by the browser executing the reference and sending the address to the crawler.

The claimed arrangement described immediately above is advantageous because the applet may have a script for producing an address for a next web page that may be executed by the applet running in the browser. The Final Office Action maintains Najork teaches "determining the address for the next page by the browser executing the reference and sending the address to the crawler," as claimed. However, Najork does not teach or suggest that a browser executes a reference that has been sent to it by a crawler in order to produce an address. Nor does Najork teach or suggest sending back to the crawler the address that results from this execution by the browser.

An example of a hypertext link that explicitly has the text of an address set out therein, as alluded to by Najork, is as follows: <a href="http://news.google.com/">. In contrast, the following hypertext link provides an example of a reference that is not so straightforward and that is "specified by a script to produce an address" so that the address for the next page is determined "by the browser executing to the reference," as stated in amended claim 1 of the present application: <a href="javascript:getMedia ('FA', '22-Sep-2004', '1', 'RM, WM');">. See Web page, http://freshair.npr.org/day_fa.jhtml?display=day&todayDate=09/21/2004. Executing this reference produces a URL such as the following:

http://www.npr.org/dmg/dmg.html?prgCode=FA&showDate=22-Sep-2004&segNum=1&NPRMediaPref=RM.

The present application says "a reference . . . may be specified by a script" because an href tag, for example, typically has a *call* to a script and not the script *itself*. The browser locates the source code for the function that is called and then executes the specified function.

Note also, a "reference" is not limited to the context of an href tag. Consider the following example snippet of HTML code:

```
<form>
<input type="button" value="GO" onclick="DoSearch()"/>
<form>
```

This snippet creates a button that says "Go." When the user presses the button the browser needs to execute the function DoSearch() in the context of the button before it can determine what URL to load. In this example also the URL to be loaded is "specified by a script," the DoSearch() script, which is not itself included in the form that produces the button.

Note also the claims subject to the present ground of rejection, like the passage of the specification set out above, state that the browser executes the *reference* instead of saying merely that the browser executes the script. In the snippet example above, the crawler needs to know what URL to load when the button is pushed. The crawler achieves this by telling the browser (via the applet) to push the button. It cannot tell the browser to just execute the JavaScript function "DoSearch()" because the browser would then not have the context in which to execute the function. See present application, page 5, lines 10 - 15 (explaining that the address "is very dependent on the context in which it is produced, that is, the history that led up to it, including the state of the server and the client browser."); see also, page 15, lines 2-8 (explaining that the crawler passes information 230 to a JavaScript execution engine 210 for generating queries to the web server 100 and that the information includes the JavaScript command that invokes script 303 when button 304 is clicked, a context object, the browser window object, and the document object associated with page 140.X in its context as it exists, loaded in browser 205).

Applicant recognizes that *executing* a reference to produce an address, as claimed, might be confused with *parsing* the reference to find an address that is explicitly set out therein. The explanation above clarifies these significant differences. Also, to make the distinction particularly clear in the claims, Applicant previously amended claim 1, to particularly point out that "a page has a reference, wherein the reference is specified by a script for producing an address for a next page" and that "such a reference is parsed from one of the web pages by the crawler program and sent to an applet running in a browser." Claims 9 and 17 were previously amended to include similar language. Further, the claims were amended to clearly tie different aspects together by stating that the address for the next page is determined by the browser "executing" the reference and sending the address to the crawler.

It should be clear from the discussion above that claims 1, 9 and 17 are patentably distinct from the teaching of Najork that the Office action relies upon for the rejection. For these reasons Applicant contends that claims 1, 9 and 17 are allowable.

Second ground of rejection for review: claims 2, 10 and 18

The references do not teach or suggest a proxy and a web site server have different IP addresses, but a resolver file indicates they are the same.

The subject claims clearly state that a proxy and a web site server have different IP addresses, but a resolver file indicates they are the same.[3] See, for example, claim 2 (stating that the browser is configured to use a certain proxy and refer to a resolver file for hostname-to-IP-address-resolution, that the resolver file indicates the IP address of the proxy *is* the IP address for the web site server, and that the IP address of the proxy *is not* the IP

[3] Because of this intentional inconsistency, the specification calls the proxy gateway a "spoof proxy." Present application, page 6, lines 13-14.

address of a web site server queried by the crawler program). Claims 10 and 18 have similar language. Appellant respectfully submits that all the limitations of the subject claims are not taught or suggested by the art relied upon and that the rejection is, therefore, improper. MPEP 2143.03 (citing *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974); *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970)).

The rejection relies on FIG. 3A in Albert, which shows a forwarding agent 302 between a client 304 and a virtual machine 310. See also Albert et al., col. 11, line 33, through col. 12, line 37. However, this does not teach or suggest an arrangement, as claimed, in which the client browser is configured to use a certain proxy and refer to a resolver file for hostname-to-IP-address-resolution, where the proxy and the web site server queried by the crawler program have *different* IP addresses but the resolver file indicates they are the *same*.

Third ground of rejection for review: claims 4, 12 and 20

The references do not teach or suggest performing particular action sequences for selecting non-hypertext-link parameters from a web page in a particular sequence, so that the queries to the web server include the selected parameters and a context arising from the particular sequence.

The subject claims state that the crawler is programmable to perform particular action sequences for selecting non-hypertext-link parameters from the at least one web page in a particular sequence, so that the queries to the web server include the selected parameters and a context arising from the particular sequence. Appellant respectfully submits that all the limitations of the subject claims are not taught or suggested by the art relied upon and that the rejection is, therefore, improper. MPEP 2143.03 (citing *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974); *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970)).

While the Final Office action does not present any substantive argument regarding these claims, a previous Office action maintained it was well-known to select non-hypertext-link parameters to dynamically generate web pages and cited Challenger for teaching about caching of such dynamically generated web pages. Office Action, dated June 22, 2005, page 6 (citing merely Challenger FIG. 1c).

Appellant acknowledges it was known to select non-hypertext-link parameters to dynamically generate web pages, and that Challenger teaches about caching of dynamically generated web pages. However, this does not teach or suggest what is claimed, which clearly sets out an "action sequence" for a programmable crawler. This may include, for example, a sequence in which "non-hypertext-link parameters," e.g., pull-down lists, buttons and the like, are "selected" from a web page. See, for example, discussion above, regarding crawler telling browser (via applet) to push a button. Further, according to the subject claims the selecting is "in a particular sequence, so that the queries to the web server include the selected parameters and a context arising from the particular sequence."

The cited teachings relied upon in the reference concern a cache manager 1 updating cached pages responsive to an application program 97 notifying cache manager 1 of a change in a record. Challenger, col. 10. As the relied upon reference states, "Caching cannot be applied to all dynamic Web pages. Some dynamic pages cause state changes . . . which must occur each time the pages are requested. Such pages cannot be cached." Challenger, col. 1, lines 60-63. Accordingly, "*For the pages that can be cached*, a need remains for a method of updating a cache when changes to underlying data which may affect the value of one or more Web pages occur." Challenger, col. 1, lines 64-66 (emphasis added). Thus the teachings of the present application, and not the cited passages of Challenger, overcome obstacles to enabling a crawler to automatically cause dynamic generation of a web page in which a query to the page causes a state change, e.g., a context arising from a particular action sequence. Accordingly, the teachings of the present application, and not the cited passages of Challenger, concern programing a crawler to perform particular action sequences for selecting non-hypertext-link parameters from a web page in a particular sequence, so that the queries to the web server include the selected parameters and *a context arising from the particular sequence*, as claimed.

Fourth ground of rejection for review: claims 7, 15 and 23

The references do not teach or suggest that a first server generated web page has included in it an operation that would cause the server to dynamically generate a second page if the first page were used to generate further requests to the server, that the operation includes a number of non-hypertext-link elements on the first page selected in a particular sequence, and that processing of the server generated web pages includes removing the operation from the first server generated web page and replacing the operation with a reference to a version of another of the server generated web pages.

Claim 1 states that a "server generated web page has included in it *an operation* that would cause the server to dynamically generate a second web page if the first page were used to generate further requests to the server" (emphasis added). The claim also states that "the operation includ[es] *a number of non-hypertext-link elements on the . . . page selected in a particular sequence*" and that the processing of the server generated web pages includes "removing the operation from the first server generated web page and *replacing the operation with a reference to a version of another of the server generated web pages* " (emphasis added). Claims 15 and 23 have similar language. Appellant respectfully submits that all these limitations of the subject claims are not taught or suggested by the art relied upon and that the rejection is, therefore, improper. MPEP 2143.03 (citing *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974); *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970)).

The Final Office Action rejects the claims in what amounts to a general allegation, without even attempting to relate all the specific aspects of the claims to specific passages of the references. Consequently, the Final Office Action misconstrues and over generalizes the claims. In particular, the Final Office Action *does not even allege* that the references specifically teach the above recited aspects of what is claimed.

Applicant submits that the references relied upon for the rejection are complex, that the Final Office action has not indicated the particular parts relied on as nearly as practicable, that pertinence of the references is not apparent, and that the pertinence as applied to the subject claims has not been explained. Applicant contends that such rejection is improper. MPEP 706, citing 37 CFR 1.104 Nature of examination (". . . When a reference is complex or shows or describes inventions other than that claimed by the applicant, the particular part relied on must be designated as nearly as practicable. The pertinence of each reference, if not apparent, must be clearly explained and each rejected claim specified.").

REQUEST FOR ACTION

For the above reasons, Appellant requests that all the pending claims of the present application be allowed and that the application be promptly passed to issuance.

Respectfully submitted,

By Anthony V S England
Anthony V.S. England
Registration No. 35,129
Attorney of Record for
IBM Corporation
Telephone: 512-477-7165
a@aengland.com

Attachments: Claims Appendix, Evidence Appendix, Related Proceedings Appendix

APPENDIX "AA" CLAIMS

1. (previously presented) A method for crawling a web site, the method comprising the steps of:

a) querying a web site server by a crawler program, wherein at least one page of the web site has a reference, wherein the reference is specified by a script to produce an address for a next page;

b) parsing such a reference from one of the web pages by the crawler program and sending the reference to an applet running in a browser; and

c) determining the address for the next page by the browser executing the reference and sending the address to the crawler.

2. (previously presented) The method of claim 1, the browser being configured to use a certain proxy and refer to a resolver file for hostname-to-IP-address-resolution, wherein the web site server has an IP address and the proxy for the browser has a certain IP address, the certain IP address of the proxy being different than the IP address of the web site server, and wherein the resolver file indicates the certain IP address of the proxy as the IP address for the web site server.

3. (original) The method of claim 2, comprising the steps of:

adding an onload attribute to one of the web pages by the proxy;

defining an event handler for the onload attribute by the proxy, wherein the event handler sets a certain variable; and

polling the certain variable by the applet to determine when the page is loaded.

4. (previously presented) The method of claim 1, wherein the crawler is programmable to perform particular action sequences for selecting non-hypertext-link parameters from the at least one web page in a particular sequence, so that the queries to the web server include the selected parameters and a context arising from the particular sequence.

APPENDIX "AA" CLAIMS

5. (previously presented) The method of claim 1, at least one of the web pages being dynamically generated by the server responsive to corresponding ones of the queries, comprising the step of:

processing the server generated web pages to generate corresponding processed versions of the web pages, so that the processed versions can be served in response to future queries, reducing dynamic generation of web pages by the server.

6. (original) The method of claim 5, wherein at least a first such server generated web page has included in it an operation that would cause the server to dynamically generate a second web page if the first page were used to generate further requests to the server, and wherein the step of processing the server generated web pages comprises the step of:

removing the operation from the first server generated web page and replacing the operation with a reference to a version of another of the server generated web pages.

7. (previously presented) A method for reducing dynamic data generation on a web site server, the method comprising the steps of:

a) querying a web site server by a crawler program responsive to references from one web page to another in the web site, wherein the queries are for causing the server to generate web pages, at least one of the web pages being dynamically generated; and

b) processing the server generated web pages to generate corresponding processed versions of the web pages, so that the processed versions can be served in response to future queries, reducing dynamic generation of web pages by the server, wherein at least a first such server generated web page has included in it an operation that would cause the server to dynamically generate a second web page if the first page were used to generate further requests to the server, the operation including a number of non-hypertext-link elements on the first page selected in a particular sequence, and wherein the step of processing the server generated web pages comprises the step of:

removing the operation from the first server generated web page and replacing the operation with a reference to a version of another of the server generated web pages.

APPENDIX "AA" CLAIMS

8. (canceled)

9. (previously presented) A computer program product for crawling a web site, wherein the computer program product resides on a computer usable medium having computer readable program code, the program code comprising:

a) first instructions for querying a web site server by a crawler program, wherein at least one page of the web site has a reference, wherein the reference is specified by a script for producing an address for a next page;

b) second instructions for parsing such a reference from one of the web pages by the crawler program and sending the reference to an applet running in a browser; and

c) third instructions for determining the address for the next page by the browser executing the reference and sending the address to the crawler.

10. (previously presented) The computer program product of claim 9, the browser being configured to use a certain proxy, and refer to a resolver file for hostname-to-IP-address-resolution, wherein the web site server has an IP address and the proxy for the browser has a certain IP address, the certain IP address of the proxy being different than the IP address of the web site server, and wherein the resolver file indicates the certain IP address of the proxy as the IP address for the web site server.

11. (original) The computer program product of claim 10, comprising:

fourth instructions for adding an onload attribute to one of the web pages by the proxy;

fifth instructions for defining an event handler for the onload attribute by the proxy, wherein the event handler sets a certain variable; and

sixth instructions for polling the certain variable by the applet to determine when the page is loaded.

APPENDIX "AA" CLAIMS

12. (previously presented) The computer program product of claim 9, wherein the first instructions comprise instructions for causing the crawler to perform particular action sequences for selecting non-hypertext-link parameters from the at least one web page in a particular sequence, so that the queries to the web server include the selected parameters and a context arising from the particular sequence.

13. (previously presented) The computer program product of claim 9, at least one of the web pages being dynamically generated by the server responsive to corresponding ones of the queries, the computer program product comprising:

instructions for processing the server generated web pages to generate corresponding processed versions of the web pages, so that the processed versions can be served in response to future queries, reducing dynamic generation of web pages by the server.

14. (original) The computer program product of claim 13, wherein at least a first such server generated web page has included in it an operation that would cause the server to dynamically generate a second web page if the first page were used to generate further requests to the server, and wherein the instructions for processing the server generated web pages to generate corresponding processed versions of the web pages comprise:

instructions for removing the operation from the first server generated web page and replacing the operation with a reference to a version of another of the server generated web pages.

15. (previously presented) A computer program product for reducing dynamic data generation on a web site server, wherein the computer program product resides on a computer usable medium having computer readable program code, the program code comprising:

first instructions for querying a web site server by a crawler program responsive to references from one web page to another in the web site, wherein the queries are for causing the server to generate web pages, at least one of the web pages being dynamically generated; and

APPENDIX "AA" CLAIMS

second instructions for processing the server generated web pages to generate corresponding processed versions of the web pages, so that the processed versions can be served in response to future queries, reducing dynamic generation of web pages by the server, wherein at least a first such server generated web page has included in it an operation that would cause the server to dynamically generate a second web page if the first page were used to generate further requests to the server, the operation including a number of non-hypertext-link elements on the first page selected in a particular sequence, and wherein the second instructions comprise:

instructions for removing the operation from the first server generated web page and replacing the operation with a reference to a version of another of the server generated web pages.

16. (canceled)

17. (previously presented) An apparatus for crawling a web site, the apparatus comprising:

a processor connected a network,

a storage device connected to the processor and the network, wherein the storage device is for storing a program for controlling the processor, and wherein the processor is operative with the program to execute a crawler program and a browser program for performing the steps of:

a) querying a web site server by the crawler, wherein at least one page of the web site has a reference, wherein the reference is specified by a script for producing an address for a next page;

b) parsing such a reference from one of the web pages and sending the reference to an applet running in a browser; and

c) determining the address for the next page by the browser executing the reference and sending the address to the crawler.

APPENDIX "AA" CLAIMS

18. (previously presented) The apparatus of claim 17, the browser being configured to use a certain proxy and refer to a resolver file for hostname-to-IP-address-resolution, wherein the web site server has an IP address and the proxy for the browser has a certain IP address, the certain IP address of the proxy being different than the IP address of the web site server, and wherein the resolver file indicates the certain IP address of the proxy as the IP address for the web site server.

19. (original) The apparatus of claim 18, wherein an onload attribute is added to one of the web pages by the proxy, and an event handler is defined for the onload attribute to set a certain variable, and wherein the processor is operative with the program for performing the step of:

polling the certain variable by the applet to determine when the page is loaded.

20. (previously presented) The apparatus of claim 17, wherein the processor is operative with the program for causing the crawler to perform particular action sequences for selecting non-hypertext-link parameters from the at least one web page in a particular sequence, so that the queries to the web server include the selected parameters and a context arising from the particular sequence.

21. (previously presented) The apparatus of claim 17, at least one of the web pages being dynamically generated by the server responsive to corresponding ones of the queries, wherein the processor is operative with the program for performing the step of:

processing the server generated web pages to generate corresponding processed versions of the web pages, so that the processed versions can be served in response to future queries, reducing dynamic generation of web pages by the server.

22. (original) The apparatus of claim 21, wherein at least a first such server generated web page has included in it an operation that would cause the server to dynamically generate a

APPENDIX "AA" CLAIMS

second web page if the first page were used to generate further requests to the server, and wherein the step of processing the server generated web pages comprises the step of:

removing the operation from the first server generated web page and replacing the operation with a reference to a version of another of the server generated web pages.

23. (previously presented) An apparatus for reducing dynamic data generation on a web site server, the apparatus comprising:

a processor connected to a network,

a storage device connected to the processor and the network, wherein the storage device is for storing a program for controlling the processor, and wherein the processor is operative with the program to execute a crawler program and a browser program for performing the steps of:

a) querying a web site server by the crawler responsive to references from one web page to another in the web site, wherein the queries are for causing the server to generate web pages, at least one of the web pages being dynamically generated; and

b) processing the server generated web pages to generate corresponding processed versions of the web pages, so that the processed versions can be served in response to future queries, reducing dynamic generation of web pages by the server, wherein at least a first such server generated web page has included in it an operation that would cause the server to dynamically generate a second web page if the first page were used to generate further requests to the server, the operation including a number of non-hypertext-link elements on the first page selected in a particular sequence, and wherein the step of processing the server generated web pages comprises the step of:

removing the operation from the first server generated web page and replacing the operation with a reference to a version of another of the server generated web pages.

24. (canceled)

APPENDIX "BB" EVIDENCE

NONE.

APPENDIX "CC" RELATED PROCEEDINGS

NONE.